RESEARCH ISSUES IN ROBOTICS

JAMES S. ALBUS

Industrial Systems Division
National Bureau of Standards
Gaithersburg, Maryland 20899

Presented at:

Society for the Advancement of
Material and Process Engineering Conference (SAMPE)
Albuquerque, NM
October 9, 1984

RESEARCH ISSUES IN ROBOTICS

James S. Albus
National Bureau of Standards
Washington, DC 20234

ABSTRACT

Six major problem areas in robotics are enumerated:

1. Kinematics, dynamics, and mobility
2. Sensors and Sensory Processing
3. Control
4. Knowledge Representation and Modeling
5. Programmable Methodology
6. Interfaces and Communications

A hierarchical robot control architecture is described which partitions the task decomposition into eight levels; four in the robot (1) servo and coordinate transformation, (2) elemental movement, (3) simple task, (4) complex task; and four in the automatic factory, (5) task sequencing (work station), (6) part batch routing (cell), (7) long range scheduling (shop), (8) process planning, product design, and management coordination (factory).

This model is used to tie together the dynamic interaction between control, sensory processing, modeling, and planning. A network architecture for robots in a small automated machine shop is used to illustrate the interface and communications issues.

KEY WORDS

Robot sensors, robot control, knowledge representation, robot programming, robot interfaces.

INTRODUCTION

Rather than attempt a comprehensive review of the state of the art in robotics, a monumental task in this rapidly moving field that encompasses so many diverse technical disciplines, I want instead to set forth a few central research topics which I believe will dominate the research community and largely occupy the attention of researchers for the rest of this century. In the course of my remarks I will cite a number of examples to illustrate the types of problems that have been, and will be, encountered in each of these research areas. But I make no claim that these examples provide a comprehensive overview of the field, or are even necessarily representative of the bulk of work currently on-going in the world today.

1. STRUCTURES

1.1 Kinematics

The first research topic that I want to address is the problem of structures. Although there are a great variety of robots on the market with many different size, shape, and form factors, much remains to be done to improve the mechanical performance of these devices.

Perhaps the most elementary problem is that of accuracy. In order to program robots off-line, it is necessary for them to be able to go to commanded coordinate points. Although the repeatability of most robots is on the order of one millimeter over the working volume (and in some cases as good as 0.1 mm.), the absolute positioning accuracy may be off as much as a centimeter. Thus it is often not possible to program a robot from an external data base, and it is not possible to transfer a program taught on one robot to another.

The present solution to the accuracy and repeatability problem is to make robot structures very stiff and rigid. Unfortunately, this means that they also tend to be massive and unwieldy. Most robots can lift only about one twentieth of their own weight. Compare that to the human arm which can lift about ten times its own weight. The difference in the strength to weight ratio is a factor of two hundred.

1.2 Dynamics

Dynamic performance is also an area where much remains to be done. Presently available robot servo systems do not adapt to the changing inertial configuration of the robot, nor do they adapt to the variety of loads that the robot must carry. The result is that robot servo systems typically are far from optimal, and often it is not possible to find any set of servo parameters that will make the robot stable over the full range of possible loads and configurations.

In the future, new mechanical designs will be needed for robots using light weight materials such as carbon filament epoxies and hollow foam-filled tubular constructions. Advanced control systems that can take advantage of light weight flexible structures are needed.

Arms that flex and bend under accelerations and loads are being investigated in the laboratory, but that work is very preliminary at this time. There is certainly nothing approaching the performance of biological arms, legs, and wings. The top slew velocity of a robot arm is typically around 40 inches per second. The top velocity that can be achieved by the human arm during a task such as throwing a base-ball is around 1500 inches per second. The difference in speed is a factor of nearly forty.

1.3 End Effectors

Much also remains to be done in robot end effectors and gripper design. Typically, robot hands consist of pinch-jaw grippers with only one degree of freedom -- open and shut. Contrast this with the human hand which has five fingers, each with four degrees of freedom. No robot hand comes close to the dexterity of the human hand.

One approach is to design interchangeable grippers and end effector tooling. But this is not without cost. Bringing sensor signals and power for control through the interchangeable interface can be a difficult process.

Another approach is to design sophisticated adaptable grippers. There have been several designs of three fingered grippers. One at the Electrotechnical Laboratory, Tsukuba, Japan, can roll a ball between its fingers or twirl a cardboard baton. But the action is slow and awkward. A similar three fingered gripper has been developed by Ken Salisbury (7), and another is under development by Steve Jackobson at the University of Utah. But the development of control algorithms for these types of grippers is in a very primitive state.

1.4 Mobility

I want to turn now briefly to the topic of mobility. Many potential robot applications require mobility. Most robots today are bolted to the floor, or to a tabletop. Small robots can reach only 20 to 50 centimeters, while larger ones can grasp objects two or three meters away. But many applications need robots which can maneuver over much larger distances. In construction tasks, such as arc welding of large structures like ships or buildings it is not practical to bring the work to the robot; the robot must go to the work, sometimes over distances of a hundred meters or more.

A really good ship building robot would be able to maneuver inside odd shaped compartments, climb over ribs and bulkheads, scale the side of the ship's hull, and weld seams several hundred feet in length. Similar mobility requirements exist in the construction of large buildings. Construction robots will need to be able to maneuver through the cluttered environment of a building site. In some cases a wheeled vehicle might be adequate, but in many applications robots will need to climb stairs, work from scaffolding, and perhaps even be suspended from cables by cranes.

Future mobile robots will be used in undersea exploration, drilling, and mining. Eventually, mobile robots will explore the moon and planets. Needless to say these applications will require significant new developments in robot mobility mechanisms.

2. SENSING

The second major problem area that I want to mention is that of sensors and processing techniques which enable robots to detect information about the state of the environment so that they can respond in an intelligent way. Robots in the automated factory will need to be able to see, feel, hear, and measure the position of objects in a number of different ways. Data from sensors must be processed, and information extracted which can be used to direct robot actions so that the robot system can successfully accomplish its task objectives in spite of uncertainties, perturbations, and unexpected conditions and events.

2.1 Machine Vision

Machine vision is the most popular research topic, and also perhaps the most difficult. The current state of the art in commercial robot vision systems is the detection and analysis of binary (black and white) silhouette images. The original work in this area was done at the Stanford Research Institute. (6) Typically, a single isolated part is photographed and the image data thresholded to produce a binary connected region. A set of features is then computed on this region. For example, the centroid, the area, the principal axis, the perimeter, and the inclusion relationships of holes can be computed. In many cases these features are sufficient to recognize an object and tell the robot where it is so that it can be picked up.

However, this method has severe limitations. For example, it cannot deal with parts that are touching or overlapping. And it does not give any information as to the three dimensional shape or position of the part.

In recent research using silhouette images, computation of the position, spacing, and orientation of features such as corners, holes, edges, and curves is performed. (4) The geometrical relationships of these features to each other can be used to characterize the image. Once this is done, these features and relationships can be compared to a model, or an ideal image of the part. If a match is detected between the features of the observed image and those of the model, then the position and orientation of the part can be computed even if it is partially hidden or obscured by touching or overlapping parts.

These binary image analysis techniques are useful primarily in situations where parts are relatively flat and lying on a known surface. It does not work well for parts that have important three dimensional contours or are stacked in piles of unknown height.

In order to deal with three dimensional relationships some form of stereo triangulation, or ranging system, must be used.

Stereo imaging has been widely researched, but the results are slow in coming. The problem is that stereo vision requires the identification of corresponding points (i.e., one must calculate which pixel in the first image is illuminated by the same point in the world as the corresponding pixel in the second image.) This is not easy to determine. It typically requires some form of cross correlation, which is a very time consuming computation.

Structured light is perhaps the most commonly used technique for simplifying the corresponding points problem. This often consists of a simple ray, or plane, of light projected on an object from one point, and viewed from another point some distance from the projector.

If the camera and light projector are mounted on the robot wrist, a single horizontal plane of light can be used to compute the distance to an object, as well as the yaw angle between the surface of the object and the robot grippers. The yaw angle is proportional to the slope of the illuminated streak.

It is in fact possible to construct a calibration chart which gives the range and x-coordinates of any illuminated point in the field or view.

If a two plane structured light system is combined with a binary image analysis program, it becomes possible to compute all six degrees of freedom of the object relative to the gripper. A pair of planes of light can measure the range, yaw, and pitch angles of a surface of an object. Binary image analysis can measure the elevation and azimuth angles of the centroid of the surface. The direction of the principal axis (or of one of the edges) can be used to compute the roll angle of the robot gripper. These measurements (range, elevation, azimuth, roll, pitch, and yaw) are the six degrees of freedom needed to control the motion of the hand of the robot relative to a surface on the object.(2)

2.2 Other Sensors

To be truly dexterous, robots need other sensors besides vision. Typically, the scanning rate for TV cameras and the processing algorithms required to extract information from vision systems are too slow for high performance servo loops. Just to scan a single image requires about 30 milliseconds. Vison processing algorithms may take several hundred milliseconds. Thus, TV camera images can be used to acquire stationary objects, or to track moving objects at a distance. But for high performance approach and gripping operations, faster acting sensors are required. For example, force servoing may require loop bandwidths greater than 100 Hertz. This corresponds to time delays of less than 10 milliseconds. Typically, proximity, force, and touch sensors can easily meet these requirements.

Work is being done at a number of different laboratories on arrays of touch sensors which enable the robot to detect the shape of the object being grasped, as well as the position of the object in the hand. At present, however, there seems to be limited utility in using large finely spaced arrays of touch sensors to recognize shape, particularly in a factory environment. Seldom does one program a robot to grasp an object by the edge such that the outline of the edge of a surface can be sensed by touch. The overall shape of an object is usually easier to measure by visual or other non-contact sensors before touch occurs, and surface orientation can be measured by as few as three tactile sensors. Of course, there are applications where sophisticated tactile shape discrimination is crucial to task performance, such as underwater where vision is obstructed by murky water. But in a factory environment such difficulties are seldom a problem.

Proximity sensors often use infra red light-emitting diodes in a variety of configurations. Sensors may measure distance as inversely proportional to reflected intensity. This requires some method of compensating for variations in reflectance of the object.

Once the object is within the grippers, beam breaking sensors can be used to detect the exact position of edges of the object. Other techniques that can be used for measuring proximity over small distances are eddy current detectors, and air pressure detectors which sense the back pressure from an air jet projected onto the surface of an object.

Acoustic sensors that measure the time of flight of an ultrasonic pulse can be used for detecting the distance to objects up to 15 feet away. The most popular commercially available acoustic ranging sensors saturate inside a few inches, so they are not useful for the terminal phase of gripping operations. However, such sensors are ideal for measuring the height of objects in a stack, or for detecting the presence of obstacles or intruders in the robot work area. Thus, they can be used for safety sensors.

3. CONTROL

The fundamental technical problem in robotics is goal-seeking, i.e. the generation and control of behavior that is successful in accomplishing a task or goal. In contrast to artificial intelligence, robotics is not primarily concerned with recognizing, classifying, naming, or understanding -- except in so far as these are required to achieve behavioral goals. The purpose of a robot control system is to accomplish commanded tasks. The purpose of sensors and sensory processing is to detect the state of the environment (i.e. the position, orientation, and spatial-temporal relationships of objects in the world) so that control signals appropriate to the task goal can be generated. This implies among other things that the processing of sensory data must be done in the context of the control problem. Because of this tight interaction between sensing and control, we will constantly intermix sensory processing in our discussion of the control system.

Most industrial robots today have no sensors, and in many cases their control system is nothing more than a memory which can store a series of points and a sequencer which can step the robot through the series of recorded points.

In the case of robots with sensors, the situation becomes more complicated. Robots with sensors require as a minimum the ability to modify the sequence of programmed points in response to sensor data. But to achieve full real-time sensory-interactive behavior, a robot must have the ability to change the actual positions of the recorded points in real time. Precomputed trajectories will not work. Trajectories must be recomputed on the fly.

Really sophisticated robot control systems need to be able to accept feedback data at a variety of levels of abstraction and have control loops with a variety of loop delays and predictive intervals. Force and velocity data used in servo loops for high speed or high precision motions can be processed and introduced into the control system with delays of no more than a few milliseconds. Vision data for detecting the position and orientation of objects to be approached typically requires hundreds of milliseconds. Processing sensory data to recognize complete objects or figure out complicated relationships between groups of objects can take seconds. Control systems that are properly organized in a hierarchical fashion so that they can accommodate a variety of sensory delays of this type are not available on any commercial robot.

In a hierarchical robot control system the bottom (or first) level of the task decomposition hierarchy is where coordinate transforms and servo computations are made. Here also all joint motions are scaled to hardware limits on velocity and force.

At the second level, elemental moves (such as <REACH TO (A)>, <LIFT>, <ORIENT ON (B)>, <MOVE TO (X)>, <RELEASE>, etc.) are decomposed into force and velocity trajectories in a convenient coordinate system. Ideally the control system will allow a coordinate frame to be defined either in the robot's work space, in the part, or in the robot's gripper.

At the third level, simple tasks (such as <FETCH (A)>, <MATE (B) TO (A)>, <LOAD TOOL (C) WITH PART (D)>, etc.) are decomposed into the set of elemental moves which can be interpreted by the second level.

Each level of the task decomposition hierarchy is serviced by a feedback processing module which extracts the information needed for control decisions at that level from the sensory data stream and from the lower level control modules. The feedback processing modules at each level detect features, recognize patterns, correlate observations against expectations, and format the results to be used in the decisions and computational procedures of the task decomposition modules at that level.

In general, sensory information at the higher levels is more abstract and requires the integration of data over longer time intervals. However, behavioral decisions at the higher levels need to be made less frequently, and therefore the greater amount of sensory processing required can be tolerated.

4. WORLD MODEL

The representation of knowledge about the world in an internal model is absolutely crucial to both the processing of sensory data and the decomposition of tasks and goals. The world model contains prior knowledge about the robot's work environment. The data in the world model may be learned (i.e., entered by storing feature parameters during a training session using a sample part), or it may be generated from a Computer Aided Design (CAD) data base which contains a geometrical representation of expected parts. In either case, the world model hierarchy contains algorithms which can compute information as to the expected shape, dimensions, and surface features of parts and tools, and may even compute their expected position and orientation at various moments in the task history. This information assists the sensory processing modules in selecting processing algorithms appropriate to the expected incoming sensory data, and in correlating observations against expectations. The sensory processing system can thereby detect the absence of expected events and measure deviations between what is observed and what is expected.

4.1 A Hierarchy of Models

At the coordinate transformation and servo level, the model generates windows or filter functions that are used to screen and track the incoming raw data stream. At the elemental move level, the model generates expected positions and orientations of specific features of parts and tools, such as edges, corners, surfaces, holes, and slots. The vision processing modules attempt to fit these models to incoming visual data. Differences between the predictions and the observations are reported back to the model, and the fitted ideal features are passed on to the next higher level as the best guess of the actual position of the features in the environment. An example of this is the two dimensional model matching work of Bolles and Cain.(4,5)

At the simple task level, the model contains knowledge of the geometrical shapes of surfaces and volumes of three dimensional objects such as parts and tools. The vision system attempts to fit the set of detected features to these surfaces and volumes. Differences between the observations and the predictions are reported back to the model, and the shifted prediction is passed on to the next higher level as the best guess as to the position and orientation of solid objects in the environment.

4.2 Obervations and Predictions

Differences between predictions and observations are measured by the sensory processing module at each level. These differences are fed back to revise the world model. New predictions generated by the revised model are then sent to the sensory processing module such that the interaction between sensory processing and world modeling is a looping, or relaxation process.

Output from the sensory processing module at each level is also used by the task decomposition hierarchy either to modify actions so as to bring sensory observations into correspondence with world model expectations, or to change the input to the world model so as to pull the expectations into correspondence with observations.

In either case, once a match is achieved between observation and expectation, recognition can be said to have been achieved. The model can then be used as the best guess of the state of the external world, and the task decomposition hierarchy can act on information contained in the model which cannot be obtained from direct observation. For example, a robot control system may use model data to reach behind an object and grasp a surface which the model predicts is there, but which is currently hidden from view. In many cases, the model can provide much more precise and noise free data about an object than can be obtained from direct measurements, which often are made under less than optimal conditions with relatively low resolution and sometimes noisy instruments. Therefore, once it has been determined that a particular model fits the object being observed, the model can provide much more complete and reliable control data than the object itself.

5. PROGRAMMING METHODS

Techniques for developing robot software must be vastly improved. Programming-byteaching is impractical for small lot production, especially for complex tasks where sensory interaction is involved.

Shop floor personnel unskilled in computers must be able to instruct robots in what to do and what to look for in making sensory decisions. The development of compilers and interpreters and other software development tools, as well as techniques for making use of knowledge of the environment derived from a number of different sensors and CAD databases are research topics that will occupy the attention of robot systems software designers for at least the next two decades.

It is not clear just yet what the characteristics of good robot programming methods will be. However, top-down structured programming techniques will surely be necessary. The real-time demands of sensory-interactive goal directed behavior imply that timing and synchronization will be a primary concern. If the control system is hierarchically structured as suggested in Section 3, there will need to be a separate programming language, or at least a separate subset of the programming language, for each level of the hierarchy. The command verbs are different at the various hierarchical levels, and the type of decisions that need to be made are also level dependent.

Nevertheless, the various levels have much in common. Each level performs a task decomposition function, and hence, much of the control system and the software which runs in it will tend to have the same logical structure.

At each level in the behavioral hierarchy, a string of commands makes up a program. This architecture implies that there is a programming language unique to each level of a hierarchical control system, and that the procedures executed by the computing modules at each level are written in a language unique to that level. Eventually, it may be necessary to have a variety of programming languages and debugging tools at each level of the sensory-control hierarchy.

The programs at each level may be written as procedures. There exist a large number of procedural robot programming languages such as VAL, AL, RAIL, RAPT, MCL, AML and others. (9) Alternatively, robot programs at each level can be represented as state graphs. (3) Of course, such a state graph can be readily transformed into a state transition table. The state transition table can then be loaded into a computing structure for execution.

At higher levels, the state transition tables are comparable to set of production rules in an expert system. Each line in the table corresponds to an IF/THEN rule. <IF (the command is such, and the state is so, and the feedback conditions are thus) / THEN (the output is whatever is stored on the right hand side of the table, and the system steps to the next state)> The addition of each node or edge to the state graph, and the corresponding lines added to the state transition table is the equivalent of the addition of a new chunk of knowledge about how to deal with a specific control situation at a particular point in a problem domain at a unique time in the task execution. This approach thus bridges the gap between servomechanisms and finite state automata at the lower levels, and expert system technologies at the upper levels. (3)

6. SYSTEM INTEGRATION

The sixth major problem area is the integration of robots into factory control systems so that many robots, machine tools, inspection devices, and materials storage, retrieval, and transportation systems can all be interconnected so as to function as a unified system.

The computing architecture shown in Figure 1 is implemented in an Automated Manufacturing Research Facility at the National Bureau of Standards.(1) It is intended as a generic system that can be applied to a wide variety of automatic manufacturing facilities. At the lowest level in this hierarchy are the individual robots, N/C machining centers, smart sensors, robot carts, conveyors, and automatic storage systems, each of which may have its own internal hierarchical control system. These individual machines are organized into work stations under the control of a work station control unit. Several work station control units are organized under, and receive input commands from a cell control unit. Several cell control units may be organized under and receive input commands from a shop control unit. At the top there is a facility control level which generates the product design, produces the manufacturing process plans, and makes the high level management decisions.

6.1 Data Bases

On the right side of the chart is shown a data base which contains the part programs for the machine tools, the part handling programs for the robots, the materials requirements, dimensions, and tolerances derived from the part design data base, and the algorithms and process plans required for routing, scheduling, tooling, and fixturing. This data is generated by a Computer-Aided-Design (CAD) system and a Computer-Aided-Process-Planning (CAPP) system. This data base is hierarchically structured so that the information required at the different hierarchical levels is readily available when needed.

On the left is a second data base which contains the current status of the factory. Each part in process in the factory has a file in this data base which contains information as to what is the position and orientation of that part, what is its stage of completion, what batch of parts it is with, and what quality control information is known. This data base is also hierarchically structured. At the equipment level, the position of each part is referenced to a particular tray or table top. At the work station level, the position of each part refers to which tray it is in. At the cell level, position refers to which work station the part is in. The feedback processors on the left scan each level of the data base and extract the information of interest to the next higher level. A management information system makes it possible for a human to query this data base at any level and determine the status of any part or job in the shop. It can also set or alter priorities on various jobs.

6.2 Interfaces

Interfaces between the many various computing modules and data bases need to be defined in some standardized way, so that large numbers of robot, machine tools, sensors, and control computers can be connected together in integrated systems.

For example, a typical workstation in a machine shop may consist of a robot, a machine tool, a work tray buffer, and several tools and sensors that the robot can manipulate. Trays of parts and tools will be delivered to the workstation by a conveyor or robot cart.

The WORKSTATION CONTROLLER will be given commands consisting of lists of operations to be performed on the parts in the trays. It is the task of the workstation controller to generate a sequence of simple task commands to the robot, the machine tool, and any other systems under its control so that the set of operations specified by its input command list are carried out in an efficient sequence. For example, the workstation controller may generate a sequence of simple task commands to the robot to setup the clamping fixtures for the first part; to the machine tool to perform the specified machining operations; to the robot to modify the clamping fixtures for the next job; etc. The planning horizon for the workstation may vary from several hours up to about a day, depending on the complexity and number of parts that are being processed.

Feedback to the workstation consists of positions of parts and relationships between various objects in order to sequence the simple task commands.

The workstation world model contains knowledge of expected tray layouts including the names of parts and their expected positions, orientations, and relationships.

The next level of the control hierarchy is the CELL CONTROLLER which is responsible for managing the production of a batch of parts within a particular group technology part family. The task of the cell is to group parts in trays and route the trays from one workstation to another. The cell generates dispatching commands to the material transport work-station to deliver the required tools, fixtures, and materials to the proper machining workstations at the appropriate times. The cell must have planning and scheduling capabilities to analyze the process plans for each part, to compute the tooling and fixturing requirements, and to produce the machining time estimates for each operation. It uses these capabilities to optimize the makeup of trays and their routing from workstation to workstation. The planning horizon for the cell will depend on the size and complexity of the batch of parts in process, but may be on the order of a week.

Feedback to the cell indicates the location and composition of trays of parts and tools and the status of activity in the workstation. This information may be derived from sensors which read coded tags on trays, or may be inferred from processed sensory input from sensors on the robot or in the workstation.

The cell world model contains information about workstation task times, and is able to predict the expected performance of various hypothetical task sequences.

The next level in the control hierarchy is the SHOP CONTROLLER which performs long term production planning and scheduling. It also manages inventory, and places orders for parts, materials, and tools. The shop control planning and scheduling functions are used to determine the material resources requirements for each cell. The shop then dynamically allocates machines and workstations to the cells as necessary to meet the production schedule.

Feedback to the shop level of control indicates the condition of machines, tools, the completion of orders, the consumption of goods, and the amount of inventory on hand.

The shop world model contains information about machine capabilities, expected tool life, and inventory levels. It is able to predict the performance of various cell configurations, and predict shortages of parts or materials in time for reordering procedures to be initiated.

The topmost level is FACILITY CONTROL. It is at this level that engineering design is performed and the process plans for maufacturing each part, and assembling each system, are generated. Here also, management information is analyzed, materials requirements planning is done, and orders are processed for maintaining inventory. Because of the very long planning horizons at this level in the control hierarchy, the activities of the facility control module are not usually considered to be part of a real-time control system. However, in a hierarchical control system, time horizons increase exponentially at each higher level. Using this concept, then, facility control activities can be integrated into the real-time control hierarchy of the total manufacturing system.

Feedback to the facility level consists of requirements for engineering changes in part design, or modifications of process plans.

The facility world model contains information about machining processes, material properties, shop processing capabilities, and expected lead times for procurements.

6.3 Interface Data Formats

One approach to the interface problem is to simply define the data elements (commands, feedback variables, status variables, sensory data parameters, etc.) which need to flow between computing modules.

These data elements can then be stored under agreed-upon names and in agreed-upon formats in the status data base. The status data base then becomes the interface between all the computing modules. At each increment of the state clock, each computing module simply reads its input variables from the status data base. It then performs its required computations, and before the end of the state clock period, writes its output back into the status data base. The status data base thus becomes the interface. An agreed upon format and protocol for the status data base then can become an interface standard.

This is analogous to the Graphics Exchange Standard (IGES). IGES is a standard data format used as the exchange medium between diverse graphics systems. (8)

The hierarchical levels described in this section correspond to well defined levels of task decomposition in the real world of manufacturing, particularly in machine shop environment. The data variables that flow between computing modules at each level correspond to physical parameters that are intrinsic to the operations being performed at those levels. There is therefore some reason to believe that it may be possible for the manufacturers and users of automated manufacturing systems to agree upon a particular set of variables to be exchanged, and a particular format for exchanging this information between computing modules. If so, then such a structure as is described here may form the basis for interface standards in the factory of the future.

7. CONCLUSION

For the most part, the six technical problem areas described above encompass profound scientific issues and engineering problems which will require much more research and development.

Yet all of the problems listed above are amenable to solution. It is only a matter of time and expenditure of resources before sensors and control systems are developed that can produce dexterous, graceful, skilled behavior in robots. Eventually, robots will be able to store and recall knowledge about the world that will enable them to behave intelligently and even to show a measure of insight regarding the spatial and temporal relationships inherent in the workplace. High order languages, computer-aided instruction, and sophisticated control systems will eventually make it possible to instruct robots using graphics generated pictures together with natural language vocabulary and syntax much as one might use in talking to a skilled worker.

As these problems are solved, robots will make ever increasing contributions to productivity improvement and the creation of real wealth.

REFERENCES

1. Albus, J.S., C.R. McLean, A.J. Barbera, M.L. Fitzgerald ,"Hierarchical Control for Robots in an Automated Factory", Proceedings of the 13th International Symposium on Industrial Robots, Chicago, 1983.

2. Albus, J., E. Kent, M. Nashman, P. Mansbach, L. Palombo, "Six-Dimensional Vision System", SPIE Vol. 336, Robot Vision 1982.

3. Albus, J.S., A.J. Barbera, and M.L. Fitzgerald, Programming a Hierarchical Control System", Proceedings of the 12th International Symposium on Industrial Robots, Paris, 1982.

4. Bolles, R.C., and R.A. Cain, "Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method", The International Journal of Robotics Research, Vol. 1,

No. 3, 1982.

5. Bolles, R.C., P.A. Horaud, M.J. Hannah, and J.A. Herson, "A System for Recognition and Location of Three-Dimensional Parts", In Machine Intelligence Research Applied to Industrial Automation, 12th Report, by D. Nitzan et al, p. 45, SRI International, Menlo Park, CA, January, 1983.

6. Rosen, C., D. Nitzan, G. Agin, G. Andeen, J. Berger, J. Eckerle, G. Gleason, J. Hill, J. Kremers, B. Meyer, W. Park, and A. Sword, "Exploratory Research in Advanced Automation", Second Report, Stanford Research Institute, Menlo Park, CA., 1974.

7. Salisbury, J.K., and Craig, J.J., "Articulated Hands: Force Control and Kinematic Issues", International Journal of Robotics Research, Vol. 1, No. 1, 1982.

8. Smith, B.M., K.M. Brauner, P.R. Kennicott, M. Liewald, J. Wellington, Initial Graphics Exchange Specification (IGES), Version 2.0, NBSIR 82-2631 (AF), National Bureau of Standards, 1983.

9. Taylor, R.H., P.D. Summers and J.M. Meyer, "AML: A Manufacturing Language", The International Journal of Robotics Research, Vol. 1, No. 3, 1982 (Bibliography references papers on other robot programming languages).

BIOGRAPHICAL SKETCH

Dr. James Albus is presently Chief of the Industrial Systems Division and Manager of the Programmable Automation Group, Center for Manufacturing Engineering, National Bureau of Standards. He is responsible for robotics and automated manufacturing systems interface standards research at NBS, and designed the control system architecture for the Automated Manufacturing Research Facility. He has written two books, "Brains, Behavior and Robotics" (McGraw-Hill 1981), and "Peoples' Capitalism: The Economics of the Robot Revolution" (New World Books 1976).

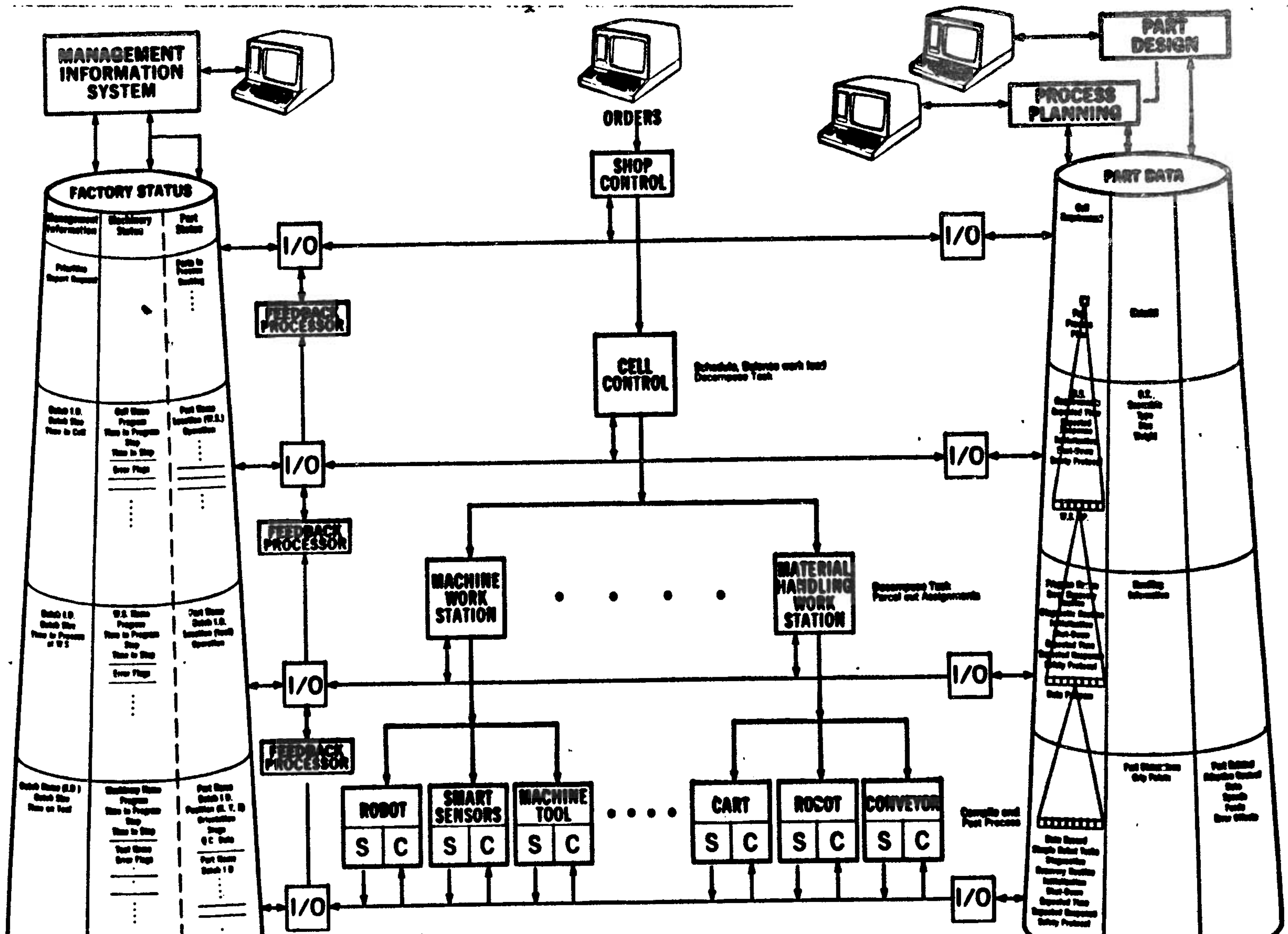

Fig. 1: A hierarchical control structure being designed at the National Bureau of Standards for controlling an automatic machine shop.